

RE: Cryptography, Encryption, Decryption Help

Source:

<http://www.tech-archive.net/Archive/PocketPC/microsoft.public.pocketpc.developer/2004-03/1228.html>

From: kladiva (*anonymous_at_discussions.microsoft.com*)

Date: 03/23/04

Date: Tue, 23 Mar 2004 06:46:10 -0800

KRoy,

Try the following, I changed the Encrypt and Decrypt functions in the Crypto.vb class to take in strings instead of Byte arrays.

'Calling Code

```
Dim encryptedtext As String = Crypto.Encrypt("212a658c-2edf-46e4-b550-ebe53b91e6b3",  
"SomeSensitiveData#23!")  
Dim cleartext As String = Crypto.Decrypt("212a658c-2edf-46e4-b550-ebe53b91e6b3", encryptedtext)
```

Replace your Encrypt and Decrypt functions with the following code:

```
Public Shared Function Encrypt(ByVal passphrase As String, ByVal cleartext As String) As String  
  
    Dim encryptedtext As String  
    Dim data() As Byte  
    data = Encoding.Unicode.GetBytes(cleartext)  
  
    ' holds encrypted data  
    Dim buffer As Byte() = Nothing  
  
    ' crypto handles  
    Dim hProv As IntPtr = IntPtr.Zero  
    Dim hKey As IntPtr = IntPtr.Zero  
  
    Try  
        ' get crypto provider, specify the provider (3rd argument)  
        ' instead of using default to ensure the same provider is  
        ' used on client and server  
        If Not WinApi.CryptAcquireContext(hProv, Nothing, WinApi.MS_DEF_PROV,  
WinApi.PROV_RSA_FULL, WinApi.CRYPT_VERIFYCONTEXT) Then  
            Failed("CryptAcquireContext")  
        End If  
  
        ' generate encryption key from passphrase  
        hKey = GetCryptoKey(hProv, passphrase)
```

microsoft.public.pocketpc.developer: RE: Cryptography, Encryption, Decryption Help

```
' determine how large of a buffer is required
' to hold the encrypted data
Dim dataLength As Integer = data.Length
Dim bufLength As Integer = data.Length

If Not WinApi.CryptEncrypt(hKey, IntPtr.Zero, True, 0, Nothing, dataLength, bufLength) Then
    Failed("CryptEncrypt")
End If

' allocate and fill buffer with encrypted data
buffer = New Byte(dataLength - 1) {}
System.Buffer.BlockCopy(data, 0, buffer, 0, data.Length)

dataLength = data.Length
bufLength = buffer.Length
If Not WinApi.CryptEncrypt(hKey, IntPtr.Zero, True, 0, buffer, dataLength, bufLength) Then
    Failed("CryptEncrypt")
End If
Finally
' release crypto handles
If Not hKey.Equals(IntPtr.Zero) Then
    WinApi.CryptDestroyKey(hKey)
End If

If Not hProv.Equals(IntPtr.Zero) Then
    WinApi.CryptReleaseContext(hProv, 0)
End If
End Try

encryptedtext = Encoding.Unicode.GetString(buffer, 0, buffer.Length)
Return encryptedtext

End Function

Public Shared Function Decrypt(ByVal passphrase As String, ByVal encryptedtext As String) As String

    Dim cleartext As String
    Dim data() As Byte
    data = Encoding.Unicode.GetBytes(encryptedtext)

    ' make a copy of the encrypted data
    Dim dataCopy As Byte() = CType(data.Clone(), Byte())

    ' holds the decrypted data
    Dim buffer As Byte() = Nothing

    ' crypto handles
    Dim hProv As IntPtr = IntPtr.Zero
    Dim hKey As IntPtr = IntPtr.Zero
```

Try

```
' get crypto provider, specify the provider (3rd argument)
' instead of using default to ensure the same provider is
' used on client and server
If Not WinApi.CryptAcquireContext(hProv, Nothing, WinApi.MS_DEF_PROV,
WinApi.PROV_RSA_FULL, WinApi.CRYPT_VERIFYCONTEXT) Then
    Failed("CryptAcquireContext")
End If

' generate encryption key from the passphrase
hKey = GetCryptoKey(hProv, passphrase)

' decrypt the data
Dim dataLength As Integer = dataCopy.Length
If Not WinApi.CryptDecrypt(hKey, IntPtr.Zero, True, 0, dataCopy, dataLength) Then
    Failed("CryptDecrypt")
End If

' copy to a buffer that is returned to the caller
' the decrypted data size might be less then
' the encrypted size
buffer = New Byte(dataLength - 1) {}
System.Buffer.BlockCopy(dataCopy, 0, buffer, 0, dataLength)
Finally
' release crypto handles
If Not hKey.Equals(IntPtr.Zero) Then
    WinApi.CryptDestroyKey(hKey)
End If

If Not hProv.Equals(IntPtr.Zero) Then
    WinApi.CryptReleaseContext(hProv, 0)
End If
End Try

cleartext = Encoding.Unicode.GetString(buffer, 0, buffer.Length)
Return cleartext
```

End Function

----- KRoy wrote: -----

I am using VS .Net 2003 to develop an VB.net Pocket PC application which encrypts connection string information for a SQL Server Database. I have copied the Crypto Class code from the Pocket PC Signiture Application Sample into my project
(<http://msdn.microsoft.com/mobility/understanding/articles/default.aspx?pull=/library/en-us/dnnetcomp/html/ppcsignatureapp.asp>).

I can not get the text I send it, to be decrypted. I

error out with an 'unhandled exception'. One thing I notice is that I get different hProv, hHash, and hKey values when running the CryptoSvc2.Decrypt function versus the CryptoSvc1.Encrypt function (this may be correct, I don't know). The following is my code:

```
Dim EncryptString As String
Dim DecryptString As String
Dim CodedArray() As Byte
Dim EncryptArray As Byte()
Dim DecryptArray As Byte()
Dim CryptoSvc1 As New PEdgeCF.Security.Crypto

CodedArray = System.Text.Encoding.ASCII.GetBytes("AnyText")
EncryptArray = CryptoSvc1.Encrypt("123456789", CodedArray)
EncryptString = System.Text.Encoding.ASCII.GetChars _
(EncryptArray)
MsgBox("Encrypted Data: " & EncryptString)

Dim CryptoSvc2 As New PEdgeCF.Security.Crypto
CodedArray = System.Text.Encoding.ASCII.GetBytes _
(EncryptString)
DecryptArray = CryptoSvc2.Decrypt("123456789", CodedArray)
DecryptString = System.Text.Encoding.ASCII.GetChars _
(DecryptArray)
MsgBox("Decrypted Data: " & DecryptString)
```

Any help would be greatly appreciated!!